* 1. 電腦的執行環境:
* 作業系統 Windows 10 專業版 64-bit
* CPU Intel Core i5 3450 @ 3.10GHz Ivy Bridge 22nm 製程
* RAM 8.00GB 雙通道-通道 DDR3 @ 665MHz (9-9-9-24)
* 編譯器 TDM-GCC 4.9.2 64-bit Release
  1. 大量資料測試之結果

**Insertion Sort**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 次數  資  料量 | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 |
| 100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 500 | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.000 | 0.000 | 0.001 | 0.001 |
| 5000 | 0.018 | 0.018 | 0.017 | 0.018 | 0.018 | 0.017 | 0.017 | 0.018 | 0.018 | 0.018 |
| 10000 | 0.071 | 0.074 | 0.072 | 0.072 | 0.075 | 0.069 | 0.122 | 0.069 | 0.084 | 0.097 |
| 50000 | 1.717 | 1.711 | 1.701 | 1.757 | 1.752 | 1.754 | 1.756 | 1.768 | 1.757 | 1.759 |
| 100000 | 6.799 | 6.857 | 6.885 | 6.820 | 6.819 | 6.812 | 6.915 | 6.787 | 6.730 | 6.737 |
| 500000 | 170.484 | 165.513 | 173.978 | 173.405 | 180.481 | 183.936 | 176.248 | 173.648 | 175.743 | 174.445 |

**Merge Sort**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 次數  資  料量 | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 |
| 100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 500 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| 1000 | 0.002 | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 | 0.002 | 0.001 | 0.002 | 0.002 |
| 5000 | 0.007 | 0.007 | 0.006 | 0.006 | 0.007 | 0.007 | 0.006 | 0.007 | 0.007 | 0.007 |
| 10000 | 0.012 | 0.012 | 0.014 | 0.012 | 0.011 | 0.011 | 0.011 | 0.012 | 0.012 | 0.012 |
| 50000 | 0.061 | 0.060 | 0.059 | 0.061 | 0.059 | 0.059 | 0.060 | 0.059 | 0.058 | 0.058 |
| 100000 | 0.123 | 0.119 | 0.119 | 0.154 | 0.119 | 0.117 | 0.116 | 0.116 | 0.116 | 0.115 |
| 500000 | 0.636 | 0.602 | 0.611 | 0.599 | 0.624 | 0.603 | 0.596 | 0.605 | 0.599 | 0.5933 |

**Quick Sort**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 次數  資  料量 | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 |
| 100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 500 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 |
| 5000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.000 | 0.001 | 0.001 | 0.001 | 0.001 |
| 10000 | 0.001 | 0.001 | 0.002 | 0.001 | 0.008 | 0.001 | 0.002 | 0.002 | 0.001 | 0.001 |
| 50000 | 0.007 | 0.008 | 0.008 | 0.008 | 0.008 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 |
| 100000 | 0.017 | 0.017 | 0.017 | 0.018 | 0.017 | 0.017 | 0.016 | 0.019 | 0.016 | 0.017 |
| 500000 | 0.109 | 0.108 | 0.110 | 0.109 | 0.109 | 0.121 | 0.111 | 0.108 | 0.110 | 0.109 |

**C qsort()**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 次數  資料量 | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 |
| 100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 500 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 5000 | 0.000 | 0.001 | 0.001 | 0.000 | 0.001 | 0.002 | 0.001 | 0.001 | 0.001 | 0.000 |
| 10000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.006 | 0.001 | 0.001 | 0.002 | 0.001 | 0.001 |
| 50000 | 0.007 | 0.007 | 0.007 | 0.007 | 0.006 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 |
| 100000 | 0.016 | 0.015 | 0.016 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.018 | 0.016 |
| 500000 | 0.071 | 0.087 | 0.075 | 0.071 | 0.071 | 0.071 | 0.072 | 0.072 | 0.072 | 0.071 |

**C++ sort()**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 次數  資  料量 | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 |
| 100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 500 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 5000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.001 | 0.001 | 0.000 | 0.000 | 0.001 |
| 10000 | 0.001 | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.002 | 0.002 |
| 50000 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 100000 | 0.021 | 0.019 | 0.017 | 0.017 | 0.017 | 0.017 | 0.017 | 0.017 | 0.017 | 0.017 |
| 500000 | 0.091 | 0.091 | 0.092 | 0.091 | 0.091 | 0.092 | 0.090 | 0.091 | 0.091 | 0.091 |

**平均時間(s)**

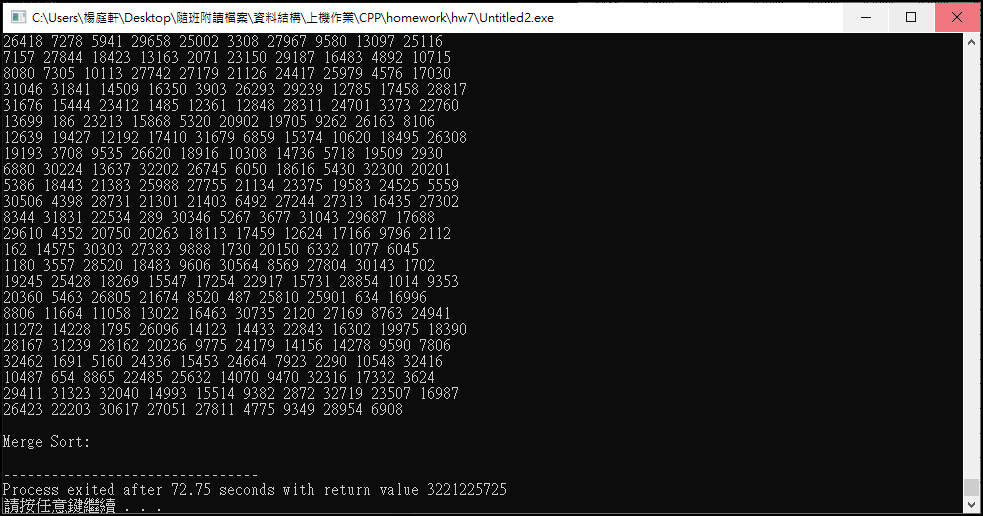
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **資料量** | **Insertion** | **Merge** | **Quick** | **C qsort** | **C++ sort** |
| 100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 500 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 |
| 1000 | 0.001 | 0.002 | 0.000 | 0.000 | 0.000 |
| 5000 | 0.018 | 0.007 | 0.001 | 0.001 | 0.000 |
| 10000 | 0.081 | 0.012 | 0.002 | 0.002 | 0.001 |
| 50000 | 1.743 | 0.059 | 0.007 | 0.007 | 0.008 |
| 100000 | 6.816 | 0.121 | 0.017 | 0.016 | 0.018 |
| 500000 | 174.788 | 0.607 | 0.110 | 0.073 | 0.091 |

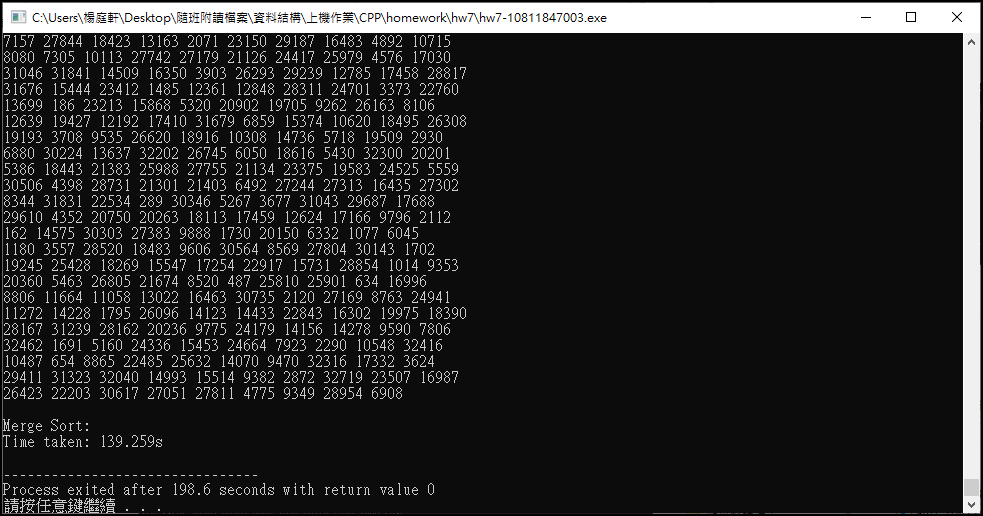
心得:

第一次做merge sort的時候，因為記憶體不夠大(8G)，所以在做50萬筆資料測試的時候，經常找不到夠大的連續記憶體空間，所以改成使用in-place的merge sort，蛋是排序時間反而變得很久。解決辦法就是使用malloc或者vector，後來我採用vector的做法來解決問題。

原本對50萬筆資料做測試，無法順利執行。

使用一般array



使用in-place merge sort

使用vector的作法

